IN THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A method of preventing live-lock in a multiprocessor system, the method comprising:

identifying a first bus transaction that is a nonmodifying transaction on [[the]] a shared resource;

identifying a second bus transaction that attempts to modify [[a]] the shared resource, wherein the second bus transaction has priority over the first bus transaction to access the shared resource;

setting a status bit to indicate that a bus transaction attempting to modify the shared resource is pending; and

retrying the first bus transaction and each subsequent nonmodifying bus transaction for the shared resource until the status bit is cleared.

2. (Previously Presented) The method of claim 1 further comprising clearing the status bit when the second bus transaction completes.

3. (Original) The method of claim 1 further comprising clearing the status bit randomly.

4. (Original) The method of claim 1 further comprising clearing the status bit at periodic intervals.

5. (Original) The method of claim 4 wherein the periodical intervals are longer than a length of time for a bus transaction to complete.

6. (Original) The method of claim 1 further comprising clearing the status bit using a pseudo-random method.

7. (Currently Amended) A method of preventing live-lock in a multiprocessor system, the method comprising:

issuing a first bus transaction to read [[the]] a cache line;

granting the cache line for the first bus transaction;

issuing a second bus transaction that attempts to modify [[a]] the cache line, wherein the second bus transaction has priority over the first bus transaction to access the cache line;

setting a status bit to indicate that a bus transaction attempting to modify the cache line is pending;

retrying the first bus transaction if the status bit is set;

reissuing the second bus transaction that attempts to modify the cache line; and

granting the cache line for the reissued second bus transaction if the status bit is set for the cache line.

8. (Previously Presented) The method of claim 7 further comprising clearing the status bit when the reissued second bus transaction complete.

9. (Original) The method of claim 7 further comprising clearing the status bit pseudo-randomly.

10. (Currently Amended) A multiprocessor computer system comprising:

a plurality of processors;

a resource shared by the plurality of processors;

at least one system bus interconnecting the shared resource and the plurality of processors;

a plurality of buffers, each one of the plurality of buffers associated with a bus transaction initiated on the at least one system bus by one of the processors; and

a status indicator associated with each one of the plurality of buffers, the status indicator being set to indicate that nonmodifying bus transactions are to be retried when a modifying bus transaction attempts to access the shared resource, wherein the modifying bus transaction has priority over the nonmodifying bus transactions to access the shared resource.

11. (Original) The multiprocessor computer system of claim 10 wherein four processors are coupled to each one of the system buses.

12. (Original) The multiprocessor computer system of claim 10 wherein the at least one system bus comprises two processor buses.

13. (Original) The multiprocessor computer system of claim 10 having four processors coupled to each one of the two processor buses.

14. (Original) The multiprocessor computer system of claim 13 further comprising an input/output bus.

15. (Currently Amended) A multiple bus, multiprocessor computer system comprising:

a plurality of processors;

a plurality of data cache memories;

a system memory shared by the plurality of processors;

at least two buses interconnecting the system memory with the plurality of data cache memories and the plurality of processors; and

a controller comprising,

a plurality of buffers, each one of the plurality of buffers associated with a bus transaction initiated on one of the buses by one of the processors; and

a status indicator associated with each one of the plurality of buffers, the status indicator being set to indicate that nonmodifying bus transactions are to be retried when a modifying bus transaction attempts to access the system memory, wherein the modifying bus transaction has priority over the nonmodifying bus transactions to access the shared resource.

16. (Original) The multiple bus, multiple processor system of claim 15 wherein each one of the at least two buses is coupled to four of the processors.

17. (Currently Amended) An integrated circuit comprising:

a bus interface to control a plurality of bus transactions;

a coherency module to maintain cache coherency for a plurality of cache lines; and

a buffer manager comprising,

a plurality of buffers, each one of the buffers to store information associated with one of the plurality of bus transactions received by the bus interface; and

a plurality of status indicators being set to indicate that nonmodifying bus transactions are to be retried when a modifying bus transaction attempts to access [[the]] one of the cache lines at least one of the status indicators associated with each one of the buffers, wherein the modifying bus transaction has priority over the nonmodifying bus transactions to access the cache lines.

18. (Original) The integrated circuit of claim 17 wherein the buffer manager further comprises logic to determine a type of bus transaction occurring on a bus.

19. (Original) The integrated circuit of claim 17 wherein the buffer manager further comprises logic to determine if two of the bus transactions are contending for a same cache line.

20. (Original) The integrated circuit of claim 17 further comprising logic to reset all of the plurality of status indicators.

21. (Original) The integrated circuit of claim 17 comprising 64 buffers and 64 status indicators.